Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: L1 and port$

Display: 10 **Documents in Display Format:** KWIC **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Wednesday, October 12, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L4	L1 and port$	1	L4
L3	L1 and schedule$	1	L3
L2	L1 and NIC	0	L2
L1	6604125.pn.	1	L1

END OF SEARCH HISTORY

Hit List

First Hit | Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 1 of 1 returned.

☐ 1. Document ID: US 6604125 B1

L3: Entry 1 of 1 File: USPT Aug 5, 2003

DOCUMENT-IDENTIFIER: US 6604125 B1
TITLE: Mechanism for enabling a thread unaware or non thread safe application to be executed safely in a multi-threaded environment

Detailed Description Text (13):
The pool ID column 202 specifies a unique ID associated with a particular thread pool. This ID allows the thread pool to be uniquely identified in the server 106. The thread type column 204 specifies the type of thread that belongs within a particular thread pool. The types of thread that can be in a system vary depending upon the system, but typical thread types include operating system scheduled threads, user or application scheduled threads, user-level operating system scheduled threads, and user-level threads. The next three columns 206, 208, 210 control the number of threads in a particular thread pool. The initial number of threads specifies the number of threads that are allocated to a thread pool when the thread pool is first created by the initialization mechanism 140 (FIG. 1). The maximum number of threads specifies the maximum number of threads that can be within a particular thread pool, and the throttle specifies the increment by which the number of threads in the thread pool may be increased.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw D

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
SCHEDULE$	0
SCHEDULE	50057
SCHEDULEABILITY	3
SCHEDULEABLE	17
SCHEDULEACKSLASH	1
SCHEDULEACTION	1
SCHEDULEACTIVELINK	1
SCHEDULEACTIVITY	1
SCHEDULEADDENTRY	1

Hit List

First Hit | Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 1 of 1 returned.

☐ 1. Document ID: US 6604125 B1

L4: Entry 1 of 1 File: USPT Aug 5, 2003

DOCUMENT-IDENTIFIER: US 6604125 B1
TITLE: Mechanism for enabling a thread unaware or non thread safe application to be executed safely in a multi-threaded environment

Detailed Description Text (23):
Once all of the thread pools are allocated, the server 106 is ready to service requests from clients 102. When a request is received (404) (at a communications port by a low level mechanism such as an operating system), a thread from the request processing mechanism thread pool is selected. This thread is assigned to and associated with the request, and is thereafter used to execute the computer code constituting the request processing mechanism 110 to process the request. Multiple requests can be processed concurrently; thus, if another request is received, then another thread is selected from the request processing mechanism thread pool, and that thread is used to execute the computer code constituting the request processing mechanism 110 to process the request.

Detailed Description Text (31):
A second function performed by the JNI_AttachThread procedure is allocation of a structure, referred to herein as a JAVA environment, to the thread being attached. This structure is stored within the private local storage of the thread (recall that threads used to execute JAVA applications have private local storage), and contains all of the resources needed by the thread to invoke the services of the JVM 116. More specifically, the JAVA environment comprises a method table portion and a local data portion. The method table contains a list of references to the methods of the JVM 116. These references may be used by the thread to invoke the services of the JVM 116 during execution of JAVA applications. The local data portion provides storage for storing local references to objects. As the services of the JVM 116 are invoked during execution of a JAVA application 132, local references to objects will be created. These local references will be stored in the local data portion of the JAVA environment. Overall, the JAVA environment provides the thread with everything that it needs to execute JAVA applications. Once the thread is registered with the JVM 116 and the JAVA environment is allocated to the thread, the thread is attached to the JVM 116 and is ready for use.

Detailed Description Text (37):
In one embodiment, the engine 126 attaches the thread to the JVM 116 by invoking the JNI_AttachThread procedure of the JVM 116. As described previously, this procedure stores the ID of the assigned thread into the internal structures of the JVM 116. In addition, it allocates and stores within the private local storage of the assigned thread a JAVA environment structure, comprising a method table portion and a local data portion. Once attached to the JVM 116, the assigned thread may be used to execute (510) one or more JAVA applications 132.

Detailed Description Text (38):
In executing the JAVA applications 132, services of the JVM 116 will be invoked. These services are invoked using the references contained in the method table of the JAVA environment. As the JVM 116 services are invoked, one or more local object references will be created. These references are stored within the local data portion of the JAVA environment. As this discussion shows, the JAVA environment provides a self-contained structure for facilitating execution of JAVA code by the thread. Execution of the JAVA applications 132 continues until the request is fully serviced. At that point, one or more response pages are generated and sent to the client 102 that submitted the request. Servicing of the request is thus completed.

Detailed Description Text (39):
After the request is serviced, it is time to return the assigned thread to the JAVA associated thread pool. Before this is done, however, some clean up operations are performed (512). Among others, these operations include clearing out the assigned thread's stack. In addition, any obsolete object references still stored in the data portion of the JAVA environment are removed. Removal of these references releases the referenced objects, which in turn enables the JVM 116 to perform garbage collection on the objects if there are no other references to them. After the clean up operations are performed, the assigned thread is returned (514) to the JAVA associated thread pool. Notice that the thread is returned to the thread pool without detaching from the JVM 116. That is, the ID of the thread is not removed from the internal structures of the JVM 116. In addition, the JAVA environment is not removed from the private local storage of the thread. Thus, for all intents and purposes, the thread is still attached to the JVM 116. That being the case, the next time that same thread is assigned from the thread pool, no reattachment will be necessary. Rather, the thread may be used as is to immediately execute JAVA applications 132. By removing the need to constantly attach and detach threads, the sticky attach mechanism of the present invention significantly decreases the overhead incurred by the server 106. This in turn increases the efficiency of the overall system.

CLAIMS:

4. The method of claim 3, where said set of indication information comprises at least a portion of a universal resource identifier (URI).

10. The apparatus of claim 9, where said set of indication information comprises at least a portion of a universal resource identifier (URI).

16. The computer readable medium of claim 15, where said set of indication information comprises at least a portion of a universal resource identifier (URI).

Term	Documents
PORT$	0
PORT	359956
PORTA	4543

PORTAABLE	1
PORTAAE	2
PORTAANKORVANKATU	1
PORTAB	2
PORTABABILITY	1
PORTABAILITY	1
PORTABALE	3
PORTABAR	1
(L1 AND PORT$).USPT.	1

There are more results than shown above. Click here to view the entire set.

Display Format: KWIC Change Format

Previous Page Next Page Go to Doc#